有向动态网络中基于模体演化的链路预测方法 *

杜　凡，刘　群

(重庆邮电大学 计算智能重庆市重点实验室，重庆 400065)

摘　要：以往传统的链路预测方法大多数针对无向网络，而实际上大多数社交网络是有向的，并且没有考虑网络中同一节点对之间的重复边以及微观演化信息，因此不能较好地解决有向动态网络中的链路预测问题。针对有向网络，将节点对之间的重复边信息转换为该节点对之间连边的权值；接着采用了基于三元组模体的演化模型，对滑动窗口中相邻时间片的模体转换概率进行统计后，采用指数加权滑动平均法对其进行时序分析得到不同模体转换概率的预测矩阵，进而使用该矩阵对网络中的链边进行预测。这不仅充分利用了网络微观演化信息，而且解决了动态网络中重复边的问题。最后对实验结果进行分析发现，在高全局聚类系数高平均度的网络中 AUC 相比 Triad Transition Matrix 方法提高了近 0.01，而相比 Common Neighbor 方法提高更多。因此，所提方法能够较好地应用网络微观演化信息进行链路预测。

关键词：时序链路预测；有向网络；模体演化；时序分析

中图分类号：TP181　　doi: 10.3969/j.issn.1001-3695.2017.11.0738

Link prediction method based on motif evolution in directed dynamic networks

Du Fan, Liu Qun

(*Chongqing Key Laboratory of Computational Intelligence*, *Chongqing University of Posts & Telecommunication*, *Chongqing* 400065, *China*)

Abstract: In the past, most of the traditional link prediction methods are oriented to the undirected network, in fact, most social networks are directional, and do not consider the duplication between the same node pair and the microscopic evolution information in the network, therefore they can not solve Link prediction in directed dynamic networks better. This paper focused on the directional network, the repeated edge information between the pair of nodes is transformed into the weight of the edge between the pair of nodes, then used the evolution model based on the triad motif, calculate the motif transformation probability matrix between the adjacent time slice in the move window, the probability matrix be analyzed by exponentially weighted moving average, and then it used the matrix to predict the chain edge in the network. This method not only makes full use of the network micro evolution information, but also solves the problem of overlapping edges in dynamic network. Experiments show that this method can get better results than Common Neighbor, Triad Transition Matrix and other methods in network with high Global Clustering Coefficient and high Average Degree．Therefore, this method can apply the network microscopic information to the link prediction better.

Key words: time series link prediction; directed network; motif evolution; time series analysis

0　引言

链路预测作为复杂网络研究中的一个重要且有趣的问题，本质上是从网络链路的微观层面解释网络结构生成的原因，进而帮助人们更好地理解网络所对应的复杂系统的结构生成和演化规律。其在计算机领域已有较深入的研究，但是其中大多数是针对静态网络，而时序链路预测方法可以利用网络的历史信息推测网络中将会产生的边。

在研究与生产环境中，链路预测有非常多的用途。例如，在生物蛋白质互助网络研究中，为降低成本，可以用链路预测算法的结果指导实验，从而降低实验成本。链路预测也可以在社交网络中用来推断两个用户之间有多大可能成为好友，从而进行推荐。另外在文献[1]所提出的对网络中错误连边的预测，对网络重构和结构功能优化也有重要的应用价值。例如对某一机场网络数据进行重建，网络中可能会有些自相矛盾的数据，链路预测就可能对其进行纠正。链路预测在复杂网络理论研究

收稿日期：2017-11-09；修回日期：2018-01-05　**基金项目：**国家自然科学基金资助资助（61572091，61075019）；重庆市自然科学基金资助项目（CSTC2014jcyjA40047）；重庆市教委研究项目（KJ1400403）；重庆邮电大学博士启动资助项目（A2014-20）

作者简介：杜凡，男（1991-），硕士研究生，主要研究方向为复杂网络、链路预测（280928338@qq.com）；刘群，女（1969-），教授，硕士，主要研究方向为复杂网络、人工智能.

方面同样具有巨大价值，它可以帮助人们认识复杂网络演化的机制，每一种网络演化机制里可能蕴涵着一种精确的链路预测方法，而每一种优秀的链路预测方法，也可能揭示了一种网络演化机制[2]。

目前比较成熟的静态链路预测方法有基于相似性的链路预测方法，如 CN(common neighbor)、Salton、Jaccard、AA（adamic adar）指标等。文献[3]提出了一种基于蚁群算法的链路预测方法。此外，还有基于最大似然估计的层次结构模型以及随机分块模型。文献[4]提出了一种基于随机分块模型的方法进行链路预测。文献[5]将边的存在与否看成边的一种属性，将链路预测问题转变为边的属性预测问题。文献[6]对复杂网络的链路可预测性问题进行了探讨，并提出了一种基于高阶路径的链路预测算法。文献[7]针对无法获取属性标签的异质网络，提出了一种包含三层图模型的学习和推理算法。针对异质网络，文献[8]还提出了一种基于节点度、共同邻居以及 Katz 三种指标的复合指标，去预测科学家合作网络的新的关系的形成。

以上方法虽然在静态网络中能取得较好的结果，它们都没有将网络的历史演化纳入考虑范围，但是现实世界中的网络大多是随时间变化的，为此，也有大量文献从网络演化的角度去进行研究。文献[9]对网络动态演化进行了量化研究。文献[10]将时序网络按时间窗口分片；然后对每个时间窗口内的网络图进行静态链路预测；最后将预测结果看做一个时间序列，并对其进行时间序列分析。文献[10]分别比较了 MA（moving average）、Av（average）、RW（random walk）、LR（linear regression）等几种时间序列预测模型，发现 LR 预测模型总体上优于其他几种模型。文献[11]提出了一种进化算法对网络中的拓扑特征和属性特征进行整合，以提高链路预测精度。文献[12]考虑链路预测面临正负样本不均衡的问题，提出了一种基于半监督学习的链路预测方法。文献[13]对一个时间片的网络图做静态链路预测后，与其后一时间片的网络图做比较，并计算出每次链路预测的误差值，最后对误差序列做时序分析，得到最终的预测误差，用其来修正最终链路预测结果，以提高链路预测精度。文献[14]提出一种整合网络的时序信息、社区结构以及节点中心性的动态网络链路预测算法。文献[15]使用基于相似性与随机游走的方法进行链路预测。文献[16]提出了一个多维时序的模型，其在 vector auto-regression(VAR)模型的基础上，在时序模型中结合拓扑矩阵，并与时间演化预测链接同时进行，可以用来预测重复的链接的发生就像预测新的链接一样。

上述这些方法各有优劣，但是都没有考虑到网络中的微观结构对网络演化的影响。模体(motif)是非常重要的一种网络微观结构，网络模体演化作为网络微观演化的一种，是网络演化分析的重要组成部分。文献[17]研究表明，网络模体的演化规律可以很大程度地揭示网络结构特征的变化。文献[18]提出了一种基于网络中的模体富集信息的聚类方法，在秀丽隐杆线虫神经元网络上应用该方法发现了由 20 个神经元组成的聚类，该聚类展示了瞬眼调节器被调控的一种途径。文献[19]挖掘三元组模体的转换概率信息得到三元组转换概率矩阵（TTM 矩阵）进行链路预测，并且取得了较好的结果。文献[20]在文献[19]的基础上，用三阶张量分解的方法计算三元组转换概率矩阵，并且在进行链路预测时考虑三元组的重要性指标。因此，结合网络模体演化对边的连接进行预测是一个可行的方向。

但是文献[19]的方法没有做充分的时序分析，而文献[20]所提方法没有考虑到网络中可能存在的重复边问题，并且该方法对无向图进行研究，而现实中的社交网络数据大多是有向图，如 facebook 评论墙网络或者 wiki 的提问回答网络，必定是由一个用户到另一个用户。为了充分利用网络微观演化信息，并且考虑网络中重复边，本文对有向动态网络进行研究，提出 MELP(motif evolution link prediction)算法进行链路预测，使用指数滑动平均的方法计算三元组转换概率矩阵，并考虑网络中边的权重与局部的结构信息。在 facebook 网络中，本文方法较 TTM 方法在时间窗口宽度为 20 时其 AUC 有 0.01 的提升，但是较 CN 方法有近 0.7 的提升。在 mathoverflow 网络中本文方法 AUC 也有提升，但是没有在 facebook 网络中显著，经分析主要是因为算法对不同拓扑结构的网络的适应性不同。

1　相关概念

1.1　问题描述

本文所解决问题可做如下描述：随时间变化的有向带权无环网络 $G=(g_1,g_2,g_3,\ldots g_T)$由 T 个时间片组成，每一个快照包含时序网络中相同时间间隔内的信息。已知时间片 g_1 到时间片 g_t 的拓扑结构，本文需要解决的就是，给出一种链路预测方法，为时间片 g_{t+1} 中的任意有序节点对赋予一个分数值，该分数值越大，则表示两节点之间越有可能产生连边。

1.2　模体理论

模体(motif)，也就是网络的基本子结构，最初在生物学里表示蛋白质网络中最基本的功能模块，这一概念也可以应用在复杂网络中。三元组是网络中最简单的模体，由三个节点组成。对于有向图，可以用 16 种三元组模体[21]进行表示，如图 1 所示，图中标注的 ID 与名称有唯一性，在本文的其他章节将会用到。

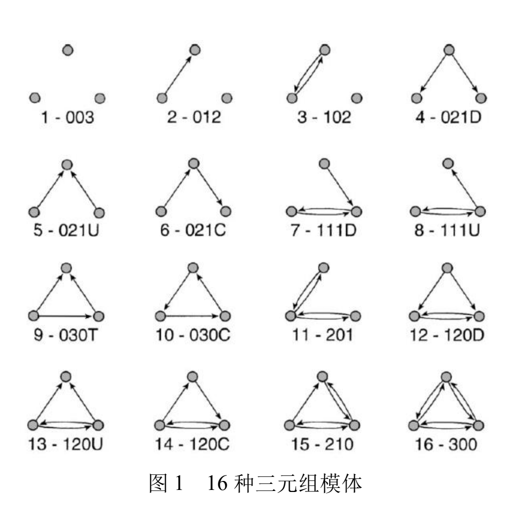


图 1　16 种三元组模体

这些不同的模体在网络的演化中有着重要的作用，本文所提方法的主要思想是利用各三元组模体之间的转换概率来进行链路预测。

对于一个网络中的任意由三个点组成的节点集，可以称为一个三元组模体 m，m 必定为图 1 中所示的 16 中模体类型之一，即为 mtf_i 的一个实例(mtf_i 表示第 i 类三元组模体)。网络的演化过程可以看做模体的不断转换。对于三个相互之间没有连边的节点(a,b,c)若在演化过程中，产生一条边 e(a,b)，这个过程可表示为

$$mtf_{003} \rightarrow mtf_{012}$$

即三元组(a,b,c)由模体类型 003 转换为模体类型 012。在此基础上，就可以描述网络的所有演化。本文对网络中所有模体类型的转换作统计，并计算各模体类型之间转换的概率 P[mtf_i->mtf_j](即为模体类型 i 转换为模体类型 j 的概率)，这个概率可以看做目标网络的一个演化特征。

例如，对于图 1 中的模体 021D 与模体 030T，通过模体转换概率计算方法经过对其在某一网络中的历史数据进行统计，发现模体 021D 到 030T 的转换概率非常高。那么对于该网络中 t 时刻一个模体为 021D 的三元组，可以知道其在 t+1 时刻，转换为 030T 的概率就会非常高，而从图 1 可知，由 021D 到 030T 需要生成一条边，那么就可以依据上述方法预测这条新连接边出现的可能性较大。

1.3　指数加权滑动平均法

指数加权滑动平均法 (exponentially weighted moving average) [22]是一种时间序列预测方法。滑动平均法 (moving average) [23]主要思想是依据一个时间序列未来可能出现的值序列与较近时期的历史观测值序列具有一定的相关性关系，进而通过取与预测期相邻的几个历史观测数据的数值平均值作为未来时间序列的预测值，得到预测结果。例如，假设数值时间序列 X=(x_1,x_2,…x_t)，需要预测 t+1 时刻的值，公式如下：

$$MA(t+1)=\frac{1}{n}\left(x_t+x_{t-1}+\ldots+x_{t-n+1}\right) \qquad (1)$$

其中：$MA(t+1)$ 表示 t+1 时刻的预测平均值。

指数加权滑动平均法是滑动平均法的改进，它既有滑动平均法的优点，又减少了数据的存储量。对于上述序列 X，使用指数加权平均法计算 t+1 时刻的预测平均值的公式如下：

$$EWMA(t+1)=\alpha x_t+(1-\alpha)EWMA(t) \qquad (2)$$

其中：$EWMA(t+1)$ 为 t+1 时刻的预测平均值；$EWMA(t)$ 为 t 时刻的预测平均值；x_t 为 t 时刻的实际值；α 为平滑系数。

1.4　连边权值

动态社交网络往往含有重复边，而本文将两节点间重复边的多少看做其关系的强弱程度。某条边重复出现的次数越多，那么这条边所代表的关系越强，所以算法开始前，可以对数据进行如下预处理：对于网络中的重复边 e_1,e_2…e_n，只保留第一次出现的边 e_1，并将该边所出现的次数 n 作为边 e_1 的权值 h，这个权值就代表着两点之间的联系的强弱程度。

2　算法设计

2.1　模体转换概率

本文主要研究基于模体演化的有向动态网络的链路预测问题，提出了一种链路预测的 MELP 算法。该算法在 TTM[19]算法的基础上，采用指数加权滑动平均法进行时序预测，并且考虑了网络中的重复边信息。

本算法首先对数据集按照某一时间跨度划分为 T 个时间片，然后对相邻两时间片之间的三元组模体进行统计，并计算模体转换概率。从图 1 可以看出，三元组模体类型总共有 16 种，因此本文定义一个 16×16 的矩阵来描述相邻时刻不同三元组模体类型的转换概率称为模体转换概率矩阵 MTM(motif transition matrix)，该矩阵行标和列标分别对应 16 种三元组模体，其中的元素表示该时刻行标对应模体转换到列标对应模体的概率，该值为

$$m_{t,i,j}=P(mtf_i[t]\rightarrow mtf_j[t+1]) \qquad (3)$$

其中：$mtf_i[t]$ 表示 t 时刻第 i 类三元组模体；$m_{t,i,j}$ 表示从 t 时刻到 t+1 时刻第 i 类三元组模体转换到第 j 类三元组模体的概率，即为对应时刻 MTM 矩阵第 i 行第 j 列的元素。转换概率矩阵计算的算法描述如算法 1 所示。

算法 1　转换概率矩阵计算

输入：t+1 时刻的图\ G，t 时刻的图 preG。

输出：16×16 的转换概率矩阵 MTM。

```
初始化 MTM;
for EACH v in G do
 vnbrs = get_neighbors(v); //vnbrs 包含与节点 v 相邻的所有节点
 for EACH u in vnbrs if u <= v then begin
   neighbors = vnbrs | get_neighbors(u) - {u, v};
//neighbors 包含与节点 u,v 相邻的所有节点
  if 边(v, u)与(u, v)同时存在于图 G
   统计模体 102 到 102，012 到 102 及 003 到 102 的转换数量;
  elseif
   统计模体 012 到 012 及 003 到 012 的转换数量;
   endif
   for each w in neighbors if u < w or (v < w < u and v not in get_neighbors(w)) then begin//get_neighbors(w)表示与 w 相邻的所有节点
        mtf_i[t] = get_triads_name(preG, v, u, w);
//get_triads_name(preG,v,u,w)获取前一时间段的图 preG 中(v,u,w)所组成的三元组的模体名称
        mtf_j[t+1] = get_triads_name(G, v, u, w);
//get_triads_name(G,v,u,w)获取当前时间段的图 G 中(v,u,w)所组成的三元组的模体名称
```

```
MTM[ mtf_i[t] ][ mtf_j[t+1] ] += 1;
    end
  end
end
```
将 MTM 矩阵每个元素除以该元素所在行的元素之和得到最终的转换概率矩阵 MTM

应用算法 1 得到所有相邻时间片之间的转换概率矩阵，依次为 MTM_1，MTM_2，…，MTM_{T-1}，如图 2 所示。

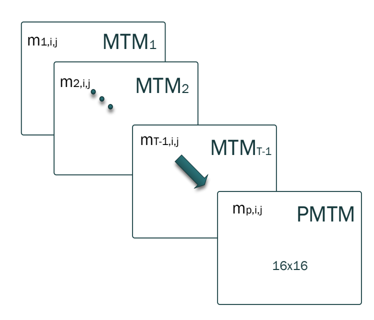


图 2　转换概率矩阵预测

本文定义一个三元组模体转换概率预测矩阵 PMTM，矩阵中元素值为

$$m_{p,i,j} = EWMA(s_{i,j}) \tag{4}$$

其中：$EWMA(s_{i,j})$ 表示对时间序列 $s_{i,j}$ 作指数加权滑动平均预测。依次取矩阵第 i 行第 j 列的值组成时间序列：

$$s_{i,j} = (m_{1,i,j}, m_{2,i,j}, \ldots, m_{T-1,i,j}) \tag{5}$$

2.2　连边分数计算

经过时间序列分析后得到三元组模体转换概率预测矩阵 PMTM。对于训练集为(T_1,T_2,…$T_{\Delta T}$)，测试集为 $T_{\Delta T+1}$ 的情况。PMTM 可以理解为通过对历史信息进行时间序列分析得到网络的 ΔT 时刻到 ΔT+1 时刻的三元组模体转换概率矩阵的预测值。

一个节点对(v,u)，有可能属于多个三元组模体。如图 3 所示，节点对(v,u)可以属于三元组(v,u,1),(v,u,2)，也可以属于(v,u,3)等，本文中只考虑与 v 或 u 相邻的节点集{1,2,3,4,5}中的元素与(v,u)所组成的三元组模体，而不考虑其他与 v 和 u 不相邻的点，如 6、7 两点。

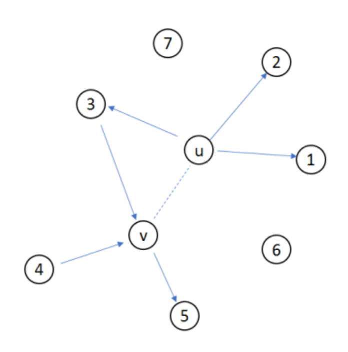


图 3　一个节点对，可能属于多个三元组模体

对于一个三元组(v,u,3)，在 v 与 u 未连边之前该三元组模体属于图 1 中的第 6 个模体，称之为 021C。如果该节点对产生连边(v,u)，则该三元组模体转换为图 1 中的第 9 个模体，命名为 030T；如果产生连边(u,v)，则转换为图 1 中的第 10 个模体，称之为 030C；如果两个方向的连边同时出现则转换为图 1 中的第 14 个模体，称之为 120C。

由于在计算节点对连边分数时，对连边分数产生影响的每一个三元组内的边具有不同的权值，而权值代表着这条边的重要程度，所以每个三元组对连边具有不同的影响力。因此，根据权值来定义连边影响因子。

$$S_{v,u,w} = \sum_{i}^{edge(v,u,w)} \sqrt{h_i} \tag{6}$$

其中：$edge(v,u,w)$ 为三元组(v,u,w)内的所有边的集合；h_i 为边 i 的权值。连边影响因子越大，说明该三元组对节点对连边贡献越大。由此，将式（6）代入式（7），可以根据矩阵 PMTM 计算得到 ΔT+1 时刻每个节点对的连边分数值：

$$score(v,u) = \sum_{w}^{nbors(v,u)} S_{v,u,w} \times PMTM\left[N_{\Delta T}, N_{\Delta T+1}\right] \tag{7}$$

其中：$N_{\Delta T}$ 为 ΔT 时刻三元组(v,u,w)的模体名；$N_{\Delta T+1}$ 为 $\Delta T+1$ 时刻三元组(v,u,w)的模体名；$nbors(v,u)$ 为节点 v,u 的邻居节点集。

3　实验数据

本文采用四个社交网络数据集对算法进行分析，这些社交网络通过对象间的互动均构成了有向网络，而且包含两节点间连边时间序列信息，所以是动态的有向网络。本文将选择一部分边作为训练集，一部分作为测试集，然后采用链路预测方法对测试集中的边进行预测，以验证本文所提方法的有效性。Facebook-wall[24]数据集是 Facebook 在美国新奥尔良地区长达 1561 天的用户留言版记录，该数据集包含三个属性，分别表示留言者、被留言者与以 UNIX 时间戳的形式保存的留言时间；sx-askubuntu[25]数据集是 askubuntu 网站用户关于 ubuntu 的问答数据，时间跨度为 2 613 天，包含三个属性，分别是回答者、提问者以及 UNIX 时间戳形式表示的回答时间；sx-mathoverflow[26]数据集是 Math Overflow 网站的用户问答数据，时间跨度为 2 350 天，该数据集所包含的属性与上一数据集基本一致。数据具体参数如表 1 所示。表中动态边数是包含重复边的数据集中的所有边数，而静态边数是指除去所有重复的边后数据集中所包含的边数。

表 1　实验数据参数

	Facebook-wall	Sx-askubuntu	Sx-mathoverflow
节点数	45813	159316	24818
动态边数	876993	964437	506550
静态边数	264004	596933	239978
时间跨度	1561day	2613day	2350day

4　实验

4.1　实验设计

为测试算法可行性，本文在上述数据集上进行实验。实验程序采用 Python 编写，运行环境为 64 位 Win10 系统。

为了全面验证本文所提方法的有效性，本节在上述三个实际数据集上对 CN、TTM[11]以及本文所提出的 MELP 算法进行对比。由于 AUC(area under the receiver operating characteristic curve) 是从整体上衡量算法的精确度，它实际上是指 ROC(receiver operating characteristic)曲线下的面积。在实际计算中，可以采用抽样比较的方法得到近似值，即每次从测试集中随机选取一条边，再从不存在的边集合中随机选择一条。如果前者分数值大于后者就加 1 分，如果两者分数值相等就加 0.5 分，所以，它也可以理解为在测试集中的分数值有比随机选择的一个不存在的边的分数值高的概率。大多数时序链路预测相关的文献中都采用 AUC 作为评价指标，因此本文实验中也采用 AUC 值作为算法评价指标。在计算 AUC 值时，采用滑动窗口的方法确定训练集与测试集：先将数据按时间间隔分片；然后确定时间窗口 ΔT，选取时间片 T_1 至 $T_{\Delta T}$ 为训练集，$T_{\Delta T+1}$ 为测试集，并计算 AUC；之后将测试集与训练集依次后移一个时间片，再次进行 AUC 计算；最终可以得到一个 AUC 序列，该序列长度为 n-ΔT，n 为时间片数量。

4.2　实验结果及分析

下面先将时间窗口宽度设置为 30。图 4 为 facebook 数据集 AUC 计算结果。可以看到 MELP 算法在 facebook 数据集的表现明显优于另外两种算法。而在 mathoverflow 数据集上 MELP 算法的表现只略优于 TTM 算法，如图 5 所示。而在 ubuntu 数据集上(图 6)，MELP 算法的表现并不如 TTM 算法但还是大大优于 CN。

为了更加全面地了解本文算法的优劣，了解不同时间窗口宽度下本文算法的表现，下面的实验分别取时间窗口宽度为 20 和 45，如图 7～12 所示。

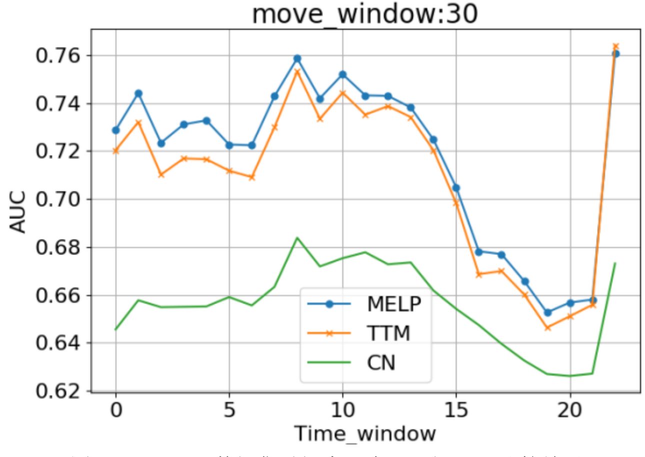


图 4　facebook 数据集时间窗口为 30 时 AUC 计算结果

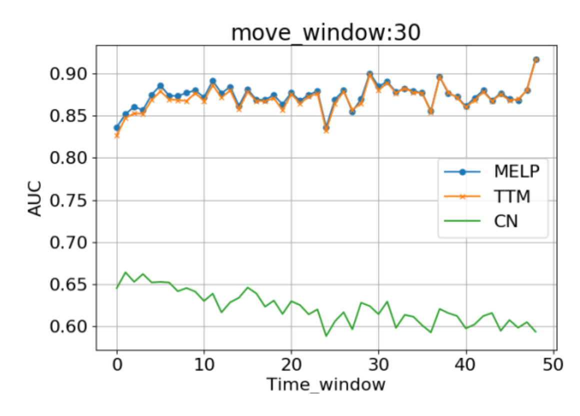


图 5　mathoverflow 数据集时间窗口为 30 时 AUC 计算结果

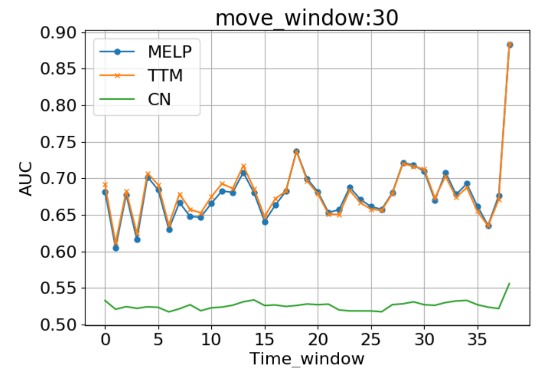


图 6　ubuntu 数据集时间窗口为 30 时 AUC 计算结果

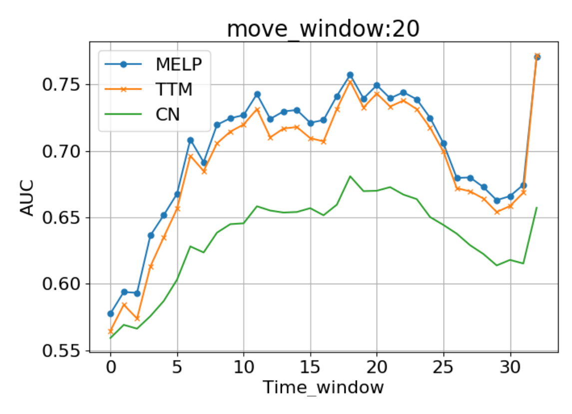


图 7　facebook 数据集时间窗口为 20 时 AUC 计算结果

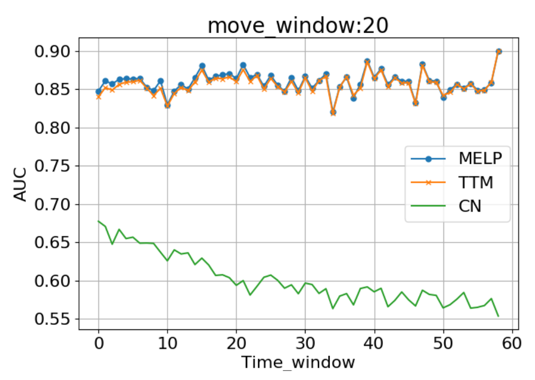


图 8　mathoverflow 数据集时间窗口为 20 时 AUC 计算结果

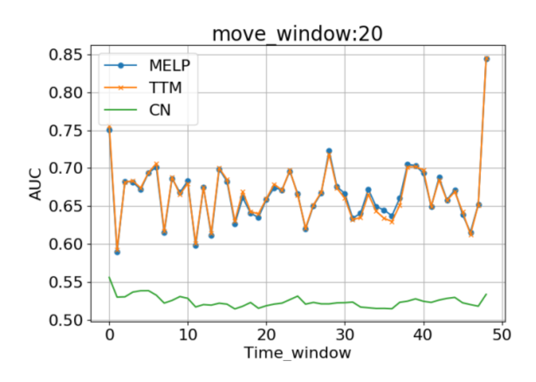


图 9　ubuntu 数据集时间窗口为 20 时 AUC 计算结果

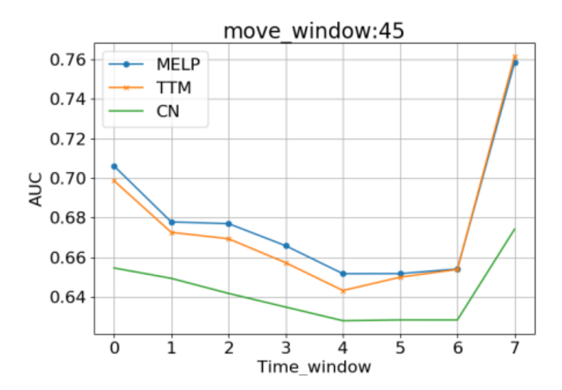


图 10　facebook 数据集时间窗口为 45 时 AUC 计算结果

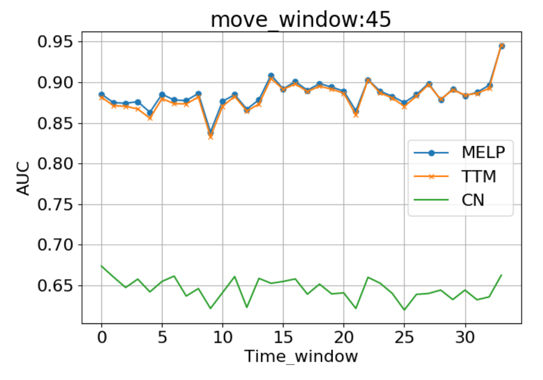


图 11　mathoverflow 数据集时间窗口为 45 时 AUC 计算结果

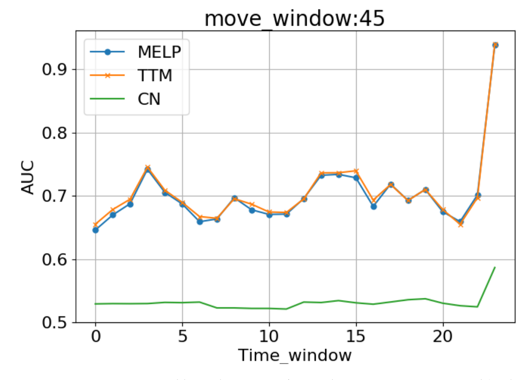


图 12　ubuntu 数据集时间窗口为 45 时 AUC 计算结果

为了更直观地看到结果，将通过滑动窗口计算得到的 AUC 序列取平均值，如表 2 所示。

表 2　AUC 平均值

data sets	time window	CN	TTM	MELP
facebook	20	0.63466013986	0.690242992424	**0.700255128205**
	30	0.656097608696	0.709551391304	**0.71752626087**
	45	0.642511057692	0.675863671875	**0.680392307692**
mathoverflow	20	0.601575749674	0.856372881356	**0.859170143416**
	30	0.622039673469	0.870864612245	**0.873813510204**
	45	0.645514819005	0.882094025735	**0.88521979638**
ubuntu	20	0.524827315542	0.666780548469	**0.667061067504**
	30	0.526036153846	**0.681565**	0.679561410256
	45	0.531397596154	**0.705387630208**	0.702055128205

从表中可看到，MELP 算法在 facebook 与 mathoverflow 数据集中都取得了更好的结果；而在 ubuntu 数据集中，只有在时间窗口宽度为 20 的情况下 MELP 算法才优与另外两种算法，所以整体来看，MELP 算法比 CN 与 TTM 更有效。

另外，从表 2 中可以发现，mathoverflow 与 ubuntu 两个数据集在所有的算法下，时间窗口宽度越大，最后结果越好。在 facebook 数据集中，时间窗口宽度为 45 时反而结果最差，而从图 4 与 7 可以看出，滑动窗口移动到靠近中间位置时 AUC 值最大。可以看出，随着窗口的滑动，窗口中包含的数据并不是稳定地有利于预测结果，当时间窗口过大时，每一个窗口中包含了过多不利于链路预测结果的数据，导致最终结果较差。而从图 5、6、8、9、11 以及图 12 可看出，对 mathoverflow 与 ubuntu 两个数据集进行链路预测的得到的 AUC 结果较为稳定，所以时间窗口宽度越大，窗口中包含越多有利于预测的信息，最后结果就越好。

接下来从网络拓扑结构的角度来分析为什么会出现上述实验结果。考虑网络的全局聚类系数(global clustering coefficient，GCC)以及平均度(average degree，AD)这两个指标。

$$\mathrm{GCC}=\frac{tri_{closed}}{tri_{closed}+tri_{open}} \tag{8}$$

$$\mathrm{AD}=\frac{2e}{n} \tag{9}$$

其中：tri_{closed} 是网络中闭合三角的个数；tri_{open} 是网络中开三角的个数。对于有向网络，从图 1 可以看到，4、5、6、7、8、11 号三元组模体是开三角，而 9、10、12、13、14、15、16 号三元组模体是闭合三角。

本文对每一个滑动窗口计算数据集的 GCC 与 AD，并对其取平均值，得到表 3。

从表 3 中可以很明显地看出，结果表现最好的 facebook 数据集在三个时间窗口大小下都具有最大的 GCC 与 AD，而表现最差的 ubuntu 数据集在三个时间窗口大小下都具有最小的 GCC 与 AD，由此可知，本文所提出算法在具有高全局聚类系数和高平均度的网络中可以得到更好的效果。

表 3 网络特征

Time Window	Data Sets	Global Clustering Coefficient	Average degree
	facebook	**0.028965**	**3.586067**
20	mathoverflow	0.009112	1.767520
	ubuntu	0.002469	1.001330
	facebook	**0.032154**	**5.386659**
30	mathoverflow	0.009959	2.625603
	ubuntu	0.002199	1.529206
	facebook	**0.031145**	**8.451428**
45	mathoverflow	0.011126	3.936553
	ubuntu	0.002078	2.280627

4.3 算法时间复杂度分析

本文提出的算法可分为两个主要部分：一是三元组模体转换概率矩阵计算，二是节点间连边分数的计算。假设网络的最大度为 d，网络节点数为 n，则模体转换概率矩阵计算时间复杂度为 $O(n*d^2)$。对于节点连边分数计算，在最坏情况下为 O(2d)，即为 O(d)。TTM 方法所用的三元组模体检测方法为遍历检测其时间复杂度为 $O(n^3)$，在节点连边分数计算上，时间复杂度与本算法一致，因此，总体来说本文方法较优。

5 结束语

时序链路预测与网络演化关系密切，将网络的微观演化规律应用到链路预测可以取得较好的结果。并且大多数链路预测方法是基于无向图的，而社交网络中某种关系的发生往往是有方向性的，基于这种关系形成的社交网络是有向网络。因此本文将动态有向网络中的三元组模体的演化应用于链路预测中，提出一种基于三元组模体演化的链路预测方法，并通过实验分析了时间窗口对实验结果的影响。

通过实验可以看出，使用动态网络中的模体转换信息来进行链路预测是可行的，并且可以得到较好的结果。在 facebook 数据集中，本文所提算法取得了明显的优势。可以证明，指数加权滑动平均法，用于对模体转换矩阵进行时间序列分析时，可以提高预测结果。在 facebook 网络中，本文方法较 TTM 方法在时间窗口宽度为 20 时其 AUC 有 0.01 的提升，但是较 CN 方法有近 0.7 的提升；在 mathoverflow 网络中本方法 AUC 也有提升，但是没有在 facebook 网络中显著；在 ubuntu 数据集中本文所提算法并没有得出更优的结果。通过对网络结构属性的分析，可以知道本算法在具有高聚类系数高平均度的网络中表现更好。在未来的工作中，本文将会进一步分析网络结构特点对算法表现的影响，将社区理论应用到本算法中，以期提高本算法在低全局聚类系数低平均度网络中的表现。此外，在以后的工作中，仍需深入分析三元组演化规律，将其与社交网络的基础理论(如三元闭包理论)相结合。最后，本文方法仍需进一步降低时间复杂度，并且在更多数据集上测试算法的有效性。

参考文献：

[1] Guimerà R, Sales-Pardo M. Missing and spurious interactions and the reconstruction of complex networks [J]. Proceedings of the National Academy of Sciences, 2009, 106 (52): 22073-22078.

[2] Dorogovtsev S N, Mendes J F F. Evolution of networks [J]. Advances in Physics, 2002, 51 (4): 1079-1187.

[3] Chen B, Chen L. A link prediction algorithm based on ant colony optimization [J]. Applied Intelligence, 2014, 41 (3): 694-708.

[4] Guimerà R, Sales-Pardo M. Missing and spurious interactions and the reconstruction of complex networks [J]. Proceedings of the National Academy of Sciences, 2009, 106 (52): 22073-22078.

[5] Taskar B, Wong M F, Abbeel P, *et al.* Link prediction in relational data [C]// Advances in Neural Information Processing Systems. 2004: 659-666.

[6] Xu Xiaoke, Xu Shuang, Zhu Y X, *et al.* Link predictability in complex networks [J]. Complex Systems & Complexity Science, 2014, 11 (1): 41-47.

[7] Kuo T T, Yan R, Huang Y Y, *et al.* Unsupervised link prediction using aggregative statistics on heterogeneous social networks [C]// Proc of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2013: 775-783.

[8] StröEle V, ZimbrãO G, Souza J M. Group and link analysis of multi-relational scientific social networks [J]. Journal of Systems and Software, 2013, 86 (7): 1819-1830.

[9] Michalski R, Bródka P, Kazienko P, *et al.* Quantifying social network dynamics [C]// Proc of the 4th International Conference on Computational Aspects of Social Networks. 2012: 69-74.

[10] Soares P R D S, Prudêncio R B C. Time series based link prediction [C]// Proc of International Joint Conference on Neural Networks. 2012: 1-7.

[11] Bliss C A, Frank M R, Danforth C M, *et al.* An evolutionary algorithm approach to linkprediction in dynamic social networks [J]. Journal of Computational Science, 2014, 5 (5): 750-764.

[12] Zeng Z, Chen K J, Zhang S, *et al.* A link prediction approach using semi-supervised learning in dynamic networks [C]// Proc of the 6th International Conference on Advanced Computational Intelligence. 2013: 276-280.

[13] Deng Z H, Lao S Y, Bai L. A temporal link prediction method based on link prediction error correction [J]. Journal of Electronics & Information Technology, 2014, 36 (2): 325-331.

[14] Ibrahim N M, Chen L. Link prediction in dynamic social networks by integrating different types of information [J]. Applied Intelligence, 2015, 42 (4): 738-750.

[15] Ahmed N M, Chen L, Wang Y, *et al.* Sampling-based algorithm for link prediction in temporal networks [J]. Information Sciences, 2016, 374: 1-14.

[16] Özcan A, Öğüdücü Ş G. Multivariate temporal Link Prediction in evolving social networks [C]// Proc of IEEE//ACIS International Conference on Computer and Information Science. 2015: 185-190.

[17] Juszczyszyn K, Musiał K, Kazienko P, *et al.* Temporal changes in local

topology of an email-based social network [J]. Computing and Informatics, 2012, 28 (6): 763-779.

[18] Benson A R, Gleich D F, Leskovec J. Higher-order organization of complex networks [J]. Science, 2016, 353 (6295): 163.

[19] Juszczyszyn K, Musiał K, Budka M. Link prediction based on subgraph evolution in dynamic social networks [C]// Proc of the 3rd IEEE International Conference on Privacy, Security, Risk and Trust. 2011: 27-34.

[20] Wang S H, Yu H T, Huang R Y, *et al.* A temporal link prediction method based on motif evolution [J]. Acta Automatica Sinica, 2016, 42 (5): 735-745.

[21] Batagelj V, Mrvar A. A subquadratic triad census algorithm for large sparse networks with small maximum degree [J]. Social networks, 2001, 23 (3): 237-243.

[22] Makridakis S G, Wheelwright S C. Forecasting methods for management [M]. [S. l.] : Wiley, 1985.

[23] Lawless J F, McLeish D L. Testing for unit roots in autoregressive-moving average models of unknown order [J]. Biometrika, 1984, 71 (3): 599-607.

[24] http://socialnetworks. mpi-sws. org/data-wosn2009. html [EB/OL].

[25] http://snap. stanford. edu/data/sx-askubuntu. html [EB/OL].

[26] http://snap. stanford. edu/data/sx-mathoverflow. html [EB/OL].